[0088] 3. Only the second stage of the fault recovery algorithm is used. Sensor estimates are assumed by each sensor to be fault free and are passed directly to the second stage without attempting to remove any faults from the estimate.

[0089] 4. An end-to-end algorithm uses both stages to remove faults and unreliable sensors.

[0090] The example examination was carried out for a system comprising five sensors. Two randomly chosen sensors undergo an un-modelled and unanticipated knock at t=50. The knock is simulated by adding a fixed value to the sensor's observations. This offset persists to the end of each run unlike the modelled faults which have all ended by t=80. The modelled fault types and their parameters are chosen randomly.

[0091] The fault start times are restrained to lie within the interval [20, 40] and the fault end times are constrained to lie within the interval [70, 80] except for the spoke fault that lasts for only one time step. Constraining the start and end times this way gives three distinct temporal regions in which we can compare our algorithms. Within the first region, up until time **20**, no fault has occurred. Sensors, which have been fault corrected using the first stage of the algorithm, are reliable until t=50 when two randomly chosen sensors are subject to a knock and become misaligned. This misalignment persisted until the end of each run.

[0092] 500 runs were gathered and the normalised standard error (NSE) over time was calculated:

$$S(t)=E[(x(t)-x(t))^T P^{-1}(t)(x(t)-x(t))].$$

[0093] For an estimator to be consistence the normalised standard error should be no greater than the cardinality of the state vector (i.e. 1 in this case). Ideally, the value of S should be close to the cardinality of the state vector indicating that the estimate covariance is not too conservative. The accuracy of the filter was also determined and this is obtained from the RMS error:

$$R(t)=\sqrt{E[(\hat{x}(t)-x(t))^T(\hat{x}(t)-x(t))]}.$$

[0094] Ideally, the RMS error is small indicating that the estimate is close to the truth. The results for these experiments are shown in FIGS. **10** and **11**, where FIG. **10** shows the result of Monte-Carlo Simulation showing the normalised standard error for the four fault recovery algorithms and FIG. **11** shows the result of Monte-Carlo Simulation showing the RMS error for the four fault recovery algorithms.

[0095] FIG. **10** demonstrates that up until t=20, when the faults set in, the algorithms have similar NSE and RMS errors. Any differences are caused by the second stage reducing the reliability of the sensors marginally. After t=20, when the modelled faults set in, the algorithms which use the first stage remain consistent and the others diverge rapidly. The RMS value, especially in the range [20, 50] is sensitive to the threshold value, β, for those algorithms which use the second stage. As β decreases the RMS value increases. The reason for this is that the second stage of the algorithm is operating throughout the run. Fault recovered sensor estimates which are very informative with values very close to the truth can be discounted by the second stage of the algorithm when they happen to be statistical outliers. This happens especially when offset faults have occurred. Algorithms that do not use the second stage would simply fuse these estimates leading to a smaller RMS error. If it were desired to guarantee a consistent estimate throughout the run, however, then it is necessary to deploy both stages of the algorithm. After t=50 the algorithm which uses both stages is the only one which remains consistent.

1. A method of estimating a state of at least one target (**101**) using a plurality of sensors (**102**), the method including:

(1) receiving (**202**) a plurality of target observations (**106**) from a respective plurality of sensors;

(2) using (**204**) the target observations to compute target state estimates;

(3) assessing (**206**) whether each of the target state estimates suffers from one of a set of modelled possible fault types;

(4) adjusting (**208**) the target state estimates to compensate for a said modelled fault type if that target state estimate is assessed to suffer from that modelled fault type;

(5) computing (**210**) a reliability value for each of the target state estimates, and

(6) fusing (**212**) the target state estimates based on the computed reliability values to produce a fused target state estimate.

2. A method according to claim **1**, wherein the steps (1), (2) and/or (3) involve a multi-hypothesis dual Kalman filter process.

3. A method according to any one of the preceding claims, further including estimating a time period during which each of the target state estimates suffered from the modelled possible fault type.

6. A method according to any one of the preceding claims, including calculating a failure hypothesis associated with each of the target state estimates, the failure hypothesis include a state covariance matrix; predictions for the target state estimate that exclude at least some of the observations; a probability value associated with the target state estimate suffering from the modelled fault type.

7. A method according to any one of the preceding claims, wherein the step of computing (**210**) a reliability value for each of the target state estimates is performed for the observation received from each sensor individually.

8. A method according to claim **7**, wherein the step of computing (**210**) the reliability of the target state estimates includes computing a distance between the target state estimate as computed using the observation from a first one of the sensors and a target state estimate as computed using the observation(s) from at least one other of the sensors.

9. A method according to claim **8**, wherein a said target state estimate is computed as being unreliable if the distance exceeds a predefined threshold value.

10. A method according to claim **9**, wherein the step of fusing (**210**) the target state estimates includes fusing the target state estimates only if the target state estimates are computed as being reliable.

11. A method according to any one of the preceding claims, wherein the fusion step (**210**) includes Kalman filter (or even covariance intersection) and mixture reduction processes.

12. A computer program product comprising a computer readable medium, having thereon computer program code means, when the program code is loaded, to make the computer execute method of a estimating a state of at least one target using a plurality of sensors according to any one of the preceding claims.

13. A system configured to estimate a state of at least one target (**101**) using a plurality of sensors (**102**), the system including: